

## **REMARKS/ARGUMENTS**

Claim 2 has been presented in an independent form. It is respectfully submitted that claim 1 be canceled without disclaimer or prejudice. New claims 21-26 have been added. However, the new claims 21-26 recite similar features as those recited in claims 10-17. Hence, no substantive claim amendment has been made. As such, it is respectfully submitted this amendment does not necessitate a new search.

The Examiner is thanked for a through search and concise remarks. In the Office Action, the Examiner has rejected claims under 35 U.S.C. 102(a) as being anticipated by "Thread-Specific Heaps for Multi-threaded programs, Bjarne Steensgaard, Microsoft Research" ("*Steensgaard*"). This rejection is traversed below.

The present application relates to creating and maintaining objects in object-oriented environments. One aspect of the invention provides, a specialized Java heap. The specialized Java heap, for example, may be used for storing Java objects with similar traits in a designated portion of the memory. Another aspect of the invention provides methods for allocating objects in a specialized heap memory. These methods can, for example, be used to create and associate objects with similar traits in a specialized heap portion or a designated portion of the memory. As will be appreciated, objects can be created and maintained more efficiently in this manner. As a result, the performance of virtual machines, especially those operating with relatively limited resources (e.g., embedded systems), is improved (see, for example, summary of the invention).

## **Rejections of Claims 2-9**

As a representative claim, claim 2 pertains to a Java heap suitable for storing Java objects. The Java heap comprises: at least one Java heap portion that is designated for storing Java objects with similar traits. It should also be noted that the at least one Java heap portion is designated to store objects of a class, and only objects of that class are stored in the at least one Java heap portion (claim 2).

It is noted that *Steensgaard* states: if it can be pre-determined that some objects allocated by a given thread never escape that thread, these objects can be allocated in a section of the heap reserved for that thread. *Steensgaard* also states that another

section of the heap can be used for objects that are shared among threads (*Steensgaard*, section 1, 2<sup>nd</sup> paragraph).

However, it is respectfully submitted that allocating objects in a section of a heap which is reserved for a thread, does not teach a Java heap portion that is designated to store objects of a class, and only objects of that class. This distinction is evident because, among other things, a thread may use a plurality of classes. Accordingly, it is respectfully submitted that claim 2 is patentable over *Steensgaard* for at least this reason alone. In addition, it is respectfully submitted that claims that are dependent on claim 2 are also patentable over *Steensgaard* for at least this reason.

Moreover, these dependent claims recite additional features that render them patentable for additional reasons. For example, claim 4 additionally recites that each of the plurality of Java heap portions be designated to store objects of the same size. It is noted that *Steensgaard* states that, if possible, small objects are allocated from a relevant heap section's last chunk (if any) of memory (*Steensgaard*, Section 3.1). However, it is respectfully submitted that allocating small objects from a relevant heap section's last chunk does not teach or suggest a plurality of heap portions that are each designated to store objects of the same size.

Furthermore, it is respectfully submitted that *Steensgaard* does not teach or suggest a Java heap portion that is designated for storing objects that do not reference other objects (claim 8). In the Office Action, the Examiner seems to be asserting that a shared object can be considered to be an object that does not reference other objects (Office Action, page 3, paragraph 8). The Applicant respectfully challenges this assertion and respectfully requests that the Examiner provides factual evidence that supports this assertion because the Applicant earnestly believes that *Steensgaard* alone cannot support the Examiner's finding that a shared object can be considered to be an object that does not reference other objects in the context of the claimed invention.

### **Rejection of Claims 10-20**

Independent claim 10 pertains to a method for creating Java objects in a heap. As such, claim 10 recites translating a Java Bytecode into one or more commands that can operate to allocate a Java object in a portion of heap that is designated for that

object, and executing the one or more commands to create the object in the designated portion of the heap. It is noted that *Steensgaard* states that instead of having a single "shared" heap section, a set of "shared" heap sections may be used by a memory manager, as well as dividing the thread-specific sections into multiple heap sections (*Steensgaard*, Section 5). However, it respectfully submitted that *Steensgaard* does not teach or suggest translating a Java Bytecode into one or more commands that can operate to allocate a Java object in a heap portion that is designated for that object. In fact, *Steensgaard* does not even address execution of a Java Bytecode to allocate a Java object. As such, it is respectfully submitted that *Steensgaard* cannot possibly teach or suggest the translation and executing features of claim 10. Accordingly, it is respectfully submitted that claim 10 and its dependent claims are patentable over *Steensgaard* for at least these reasons. It should also be noted that the dependent claims 11-17 recites several additional features that render them patentable over *Steensgaard* for additional reasons. These additional features, for example, include marking and compiling features respectively recited in claims 11 and 14. Contrary to the Examiner's assertion, it is respectfully submitted that *Steensgaard* does not teach or suggest these features.

Although independent claim 18 pertains to a computer readable media, it recites similar features as recited in claim 10. Accordingly, it is respectfully submitted that claims 18 and its dependent claims are also patentable over *Steensgaard* for similar reasons as discussed above with respect to claim 10.

Based on the foregoing, it is submitted that claims are patentably distinct over the cited art of record. Additional limitations recited in the independent claims or the dependent claims are not further discussed because the limitations discussed above are sufficient to distinguish the claimed invention from the cited art. Accordingly, Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to

Deposit Account No. 500388 (Order No. SUN1P819). Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,  
BEYER WEAVER & THOMAS, LLP



R. Mahboubian  
Reg. No. 44,890

P.O. Box 778  
Berkeley, CA 94704-0778  
(650) 961-8300